# CSNS-II 加速器控制网络管理系统设计与实现

薛康佳 [1,2] 张玉亮 [1,2,3] 朱鹏 [1,2,3] 吴煊 [1,2,3] 王林 [1,2,3]

李明涛 [1,2,3] 何泳成 [1,2,3] 程司农 [1,2] 彭娜 [1,2]

1（中国科学院高能物理研究所 北京 100049）

1（散裂中子源科学中心 广东 东莞 523803）

2（中国科学院大学 北京 100049）

**摘要** 随着基于粒子加速器的大科学装置的规模和复杂度不断提升，其控制网络面临着设备数量激增、安全管控困难、维护效率低下等挑战。针对这些问题，设计并开发了一套面向大型加速器的控制网络管理系统。该系统实现了控制网络 IP 地址统一管理、网络动态信息自动采集以及网络接入控制三大核心功能。通过集中申请与审批机制避免 IP 冲突问题；基于交换机运行数据实现设备在线状态实时监控和物理位置精确定位；采用 IP 与端口绑定方案确保控制网络接入安全。该系统采用 Web 架构进行设计和实现，前端基于 Vue.js 框架开发，后端采用 Node.js 与 Python 混合技术栈，数据存储选用 MongoDB 数据库。该系统已在 CSNS 加速器控制网络中成功部署并稳定运行，有效解决了网络管理中的安全隐患和维护难题，为 CSNS-II 网络管理奠定了基础。

**关键词** 控制网络；控制系统；加速器；网络管理；中国散裂中子源

**中图分类号** TL503.6

## Design and implementation of CSNS-II accelerator control network management system

XUE Kangjia [1,2] ZHANG Yuliang [1,2,3] ZHU Peng [1,2,3] WU Xuan [1,2,3] WANG Lin [1,2,3]

LI Mingtao [1,2,3] HE Yongcheng [1,2,3] Cheng Sinong [1,2] Peng Na [1,2]

1(*Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China*)

2(*Spallation Neutron Source Science Center, Dongguan Guangdong 523808, China*)

3(*University of Chinese Academy of Sciences, Beijing 100049, China*)

**Abstract** **[Background]**: The increasing scale and complexity of large scientific facilities have led to significant challenges in control networks, including a proliferation of connected devices, security management difficulties, and inefficient maintenance processes. **[Purpose]**: To address these challenges, a comprehensive control network management system was designed and implemented, specifically tailored for large-scale accelerator facilities. **[Methods]**: The developed system integrates three core functionalities: unified IP address management, dynamic network information acquisition, and network access control. A centralized application and approval mechanism was implemented to eliminate IP conflicts. Real-time monitoring of device status and precise physical location identification were achieved through continuous collection of switch operation data. Network security was enhanced through an IP-port binding scheme that strictly controls access to the control network. The system architecture employs a web-based approach with a Vue.js frontend framework, a hybrid Node.js and Python backend technology stack, and MongoDB for data persistence. **[Results]**: The system has been successfully deployed and is running stably in the CSNS accelerator control network. It effectively resolves security vulnerabilities and maintenance

中国科学院青年创新促进会(Y9291420K2)资助

第一作者：薛康佳，男，1991 年出生，从事加速器控制系统研究

通讯作者：薛康佳，E-mail: xuekj@ihep.ac.cn

收稿日期：20XX-00-00，修回日期：20XX-00-00

challenges in network management. **[Conclusions]**: The implemented management system significantly enhances both the operational security and maintenance efficiency of accelerator control networks. It establishes a robust foundation for CSNS-II network management and offers a promising solution for improving control network administration across various large-scale scientific facilities.

**Key words** Control network; Control system; Accelerator; Network management; CSNS

随着加速器规模和复杂度的不断提升，控制网络管理与维护的挑战随之增加。以中国散裂中子源(CSNS:China Spallation Neutron Source)加速器为例，其控制网络目前在线设备已达 650 个，而 CSNS-II 建成后预计将超过 1500 个网络设备。在设备数量急剧增长的背景下，传统的网络管理模式已难以满足高效、安全运行的需求，因此亟需建立一套完善的网络管理系统以预防潜在风险。

当前 CSNS 加速器控制网络管理面临多方面的挑战。在安全层面，由于缺乏统一的接入审批机制，任意人员可随意将设备接入网络，增加了安全隐患；控制系统给各系统分配 IP 地址段后，各系统自行分配所属设备的 IP 地址，易导致 IP 冲突，从而影响网络稳定性。在维护层面，管理人员难以快速获取 IP 所对应设备的详细信息，且缺乏 IP 与网络端口关联的数据，这增加了故障排查和网络维护的难度。

随着 CSNS-II 工程规模的扩大及网络设备数量的显著增加，目前的网络管理方法面临的挑战将进一步加剧。针对这些问题，本文提出并实现了 CSNS-II 加速器控制网络管理系统。该系统通过集中申请与审批机制避免 IP 冲突问题；基于交换机运行数据实现设备在线状态实时监控和物理位置精确定位；采用 IP 与端口绑定方案确保控制网络接入安全。这些功能的实现旨在解决当前 CSNS 加速器控制网络管理面临的挑战，为 CSNS-II 的高效、安全运行提供有力支持。

## 1 系统总体设计

### 1.1 CSNS-II 加速器控制网络简介

CSNS-II 加速器控制系统是一个复杂的大型分布式控制系统，其主要任务是使操作员能够通过人机接口装置操作系统设备，按照 CSNS-II 设计的物理要求进行实时控制，对高速运行的质子轨道进行精确定位和控制。为此控制系统将对分布在直线加速器、输运线和储存环等多个区域的设备进行控制[1,2]，这些设备的控制均由连接以太网的计算机进行实时监控，使之协同工作，实现对设备的控制。

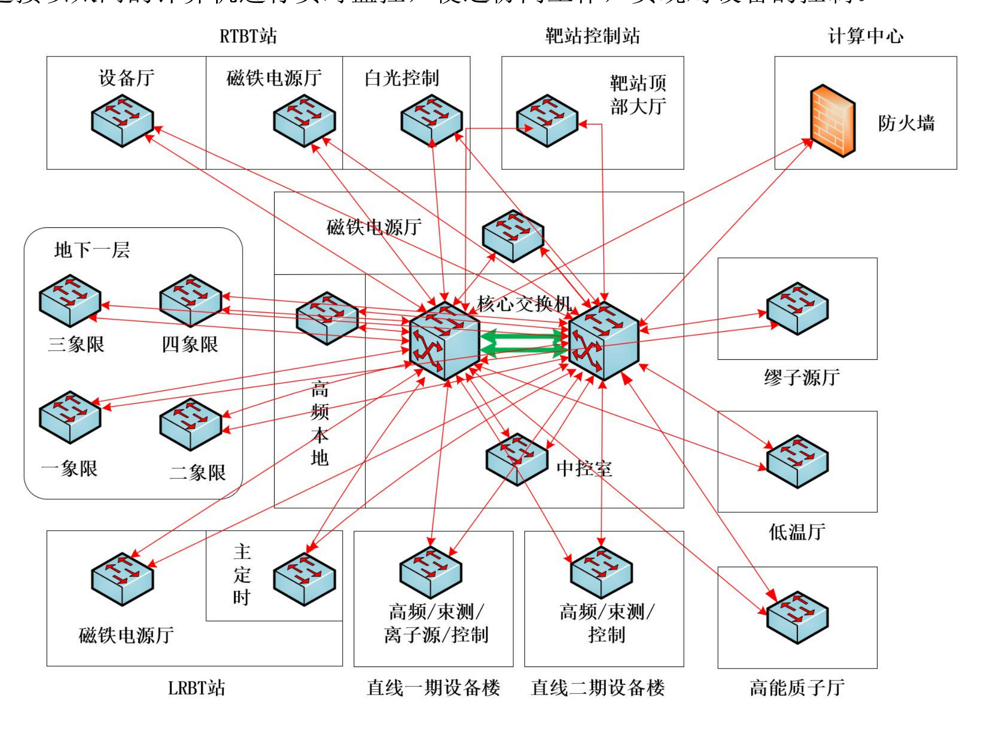



**图 1** CSNS-II 加速器控制网络结构
**Fig.1** Structure of CSNS-II accelerator control network

CSNS-II 控制网络采用星型二层结构，网络结构如图 1 所示。核心层位于中央控制室服务器机房内，部署 2 台核心交换机，通过高速光纤链路实现冗余互联，采用 VRRP 协议实现热备，确保系统的高可用性，当一台核心交换机故障时，另一台能够无缝接管数据流量[3,4]。接入层分布在各个设备厅，负责连接终端设备，CSNS-II 建成后预计将有 50 台接入交换机和超过 1500 个网络设备。每台接入层交换机都采用 10G 双上联方式连接至核心层交换机，形成全冗余上联链路。这种双上联设计使得即使一条上联链路或一台核心交换机故障，数据仍能通过另一条路径正常传输，确保了整个控制网络运行的稳定。核心与接入层之间的 10G 链路采用链路聚合技术，不仅提供了 20G 的总带宽能力，还实现了流量的智能分担。考虑到接入设备数量庞大，控制网络划分为多个虚拟局域网 (VLAN)，并在每个 VLAN 内细分 IP 段分配给各子系统。这种 VLAN 划分方案有效隔离了不同子系统的广播流量，减少了网络拥塞风险，同时提高了网络安全性[5,6]。

### 1.2　网络管理系统总体设计

尽管市面上存在众多商业网络管理软件，但这些软件主要针对常规园区网络管理设计，无法满足加速器网络管理的特殊需求。例如与园区办公计算机相比，加速器的网络设备如控制、电源、高频、真空等大部分系统的设备不具备自行申请网络接入的条件，且由于加速器需要连续运行，不适合采用绑定 MAC 地址的方案，这将导致设备故障时更换困难，可能影响加速器运行效率。另外，相较于园区网络通常仅需记录使用人和设备名称，控制网络需要对接入设备的信息有更全面的掌握。

针对加速器的特殊需求，本文设计的网络管理系统旨在不影响加速器运行的前提下，提升控制网络的安全性和维护效率。在安全性方面，基于加速器的设备特点，设计了基于 IP 绑定的接入方案，当设备更换时只需配置相同的 IP 地址，无需再次申请接入网络，从而确保不影响加速器运行。在维护效率方面，设计了控制网络 IP 地址统一管理方案，同时记录设备的详细信息，并将设备静态信息与网络动态信息相结合，实现快速定位设备位置和监控在线状态等功能。

网络管理系统采用了三层架构设计，包括数据呈现层、数据处理层和设备交互层，总体架构如图 2 所示。这种分层架构不仅提高了系统的模块化程度，也增强了其可扩展性和维护性。系统的实现主要涉及 Web 应用程序开发和网络管理技术，总体技术方案如图 3 所示。

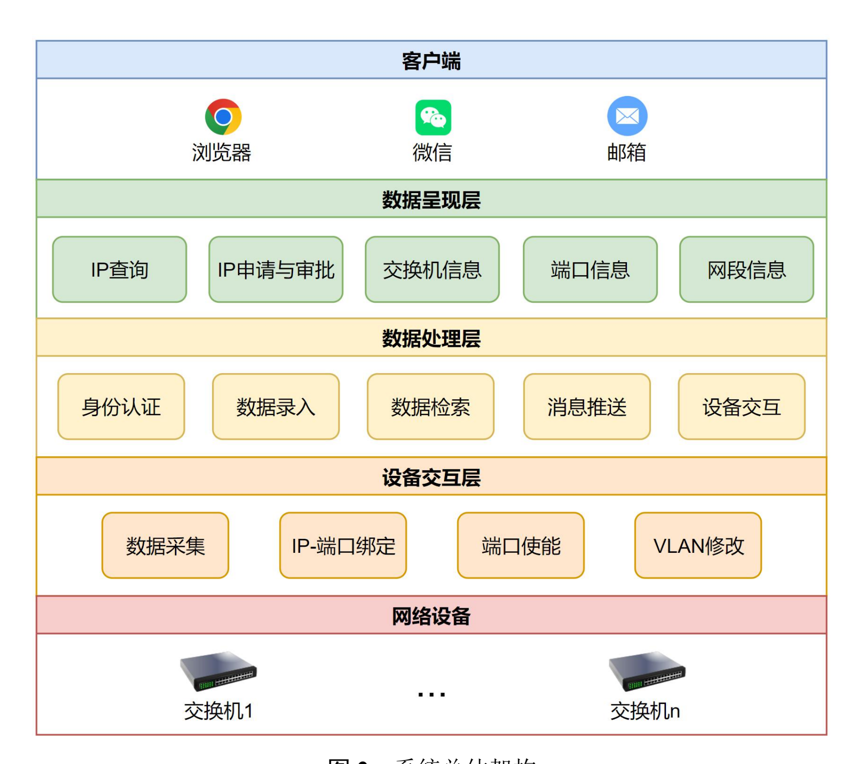



**图 2**　系统总体架构
**Fig.2**　Overall system architecture

数据呈现层采用 Web 架构作为用户交互接口，并结合微信和邮箱推送系统消息。相较传统桌面应用，Web 应用易于部署和更新，且可以直接在不同的操作系统和设备上运行，用户仅需标准 Web 浏览器和网络连接即可随时访问。系统支持通过浏览器查询 IP 信息、填写及审批 IP 地址使用申请、查看交换机信息、交换机端口信息以及查看网段信息。在 Web 前端技术选型方面，系统采用了 Vue.js 框架作为主要技术栈。Vue.js 是一种渐进式的 JavaScript 框架，凭借其轻量化、高性能以及组件化开发模式的特点而成为了目前主流的前端开发框架之一[7]。为了提高开发效率和用户体验，系统集成 PrimeVue 组件库作为 UI 实现方案。PrimeVue 是一个专为 Vue.js 设计的 UI 组件库，提供了功能丰富、可定制的组件，简化了复杂 UI 的实现过程，同时保证了系统界面的一致性和专业性[8]。

数据处理层设计采用微服务架构，提供身份认证、数据录入、数据检索、消息推送以及调用设备交互层的服务。系统采用了基于 Node.js 运行环境的 Express 框架和 Python 运行环境的 Flask 框架。Node.js 作为一种高性能、事件驱动的 JavaScript 运行时，能够有效支持高并发、低延迟的网络服务[9]。Python 在网络编程和设备交互方面拥有丰富的类库和成熟的生态系统，能够显著简化与交换机数据传输和交互的开发流程，提高开发效率[10]。采用异构语言的微服务架构发挥了不同编程语言的优势，既保证了系统的整体性能，又为未来的技术扩展预留了空间。

设备交互层由多个 Python 语言编写的设备交互模块组成，主要实现网络数据采集、IP 和端口绑定操作、端口使能状态修改以及端口 VLAN ID 变更等功能。网络数据采集模块根据采集的数据类型设置了不同的定时执行周期，而设备操作相关模块则通过数据处理层的被动调用来执行。

在数据存储方面，系统采用 MongoDB 数据库，这是一种 NoSQL（非关系型）数据库[11]，以其高性能、灵活的数据模型和良好的横向扩展能力而在现代 Web 系统开发中得到广泛应用。与传统关系型数据库采用预定义的表结构不同，MongoDB 的文档存储模型基于 JSON 格式的 BSON 文档[12]，这种模式能够更灵活地适应多样化和动态变化的数据需求，高度契合本系统的需求特性。

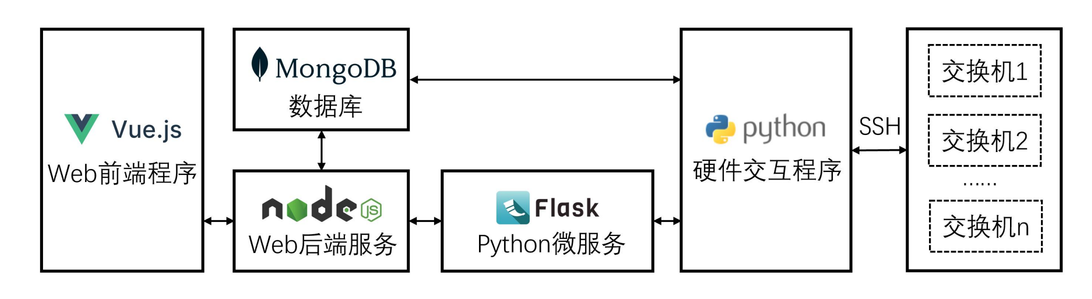



**图 3**　系统技术方案
**Fig.3**　Technical scheme of the system

## 2　控制网络 IP 地址管理

### 2.1　控制网络 IP 管理方案设计

CSNS 加速器的网络设备采用静态 IP 的配置方式。在此前的管理方案中，管理员将可分配的 IP 段分配给各子系统后，由相应责任人自行分配 IP 地址。这种方式存在两个主要问题：首先，网络管理员无法统一掌握 IP 信息，导致出现网络问题时难以直接定位设备；其次，由于各子系统管理上的差异，长期运行后造成 IP 信息记录不全，增加了 IP 冲突的风险。

为解决上述问题，本文实现了控制网络 IP 地址的统一管理机制，主要包括 IP 地址分配、信息查询、使用申请和审批等功能。

管理员预先在系统中分别导入网段、交换机、端口及可分配 IP 的列表，这些列表包含了预设的网络信息，例如划分的网段名称和对应的 VLAN ID，每台交换机的名称、管理 IP 及所在位置，每个端口的线缆编号等。

用户新增设备接入控制网络前，需在 IP 总览页面选择未分配的 IP 地址提交使用申请，经管理员审批

后方可接入。申请过程中，用户可查看每个 IP 地址预分配的子系统，从而保持各子系统使用统一分配的网段及 IP 段。IP 地址申请表单支持填写设备名称、管理信息、主机名、用途、使用的子系统、设备厂商、型号以及连接的交换机名称和端口等信息，全面涵盖网络管理所需的设备信息。其中设备管理信息仅申请人可见，其他信息则所有人员均可查看。

若填写的信息不满足网络管理要求，管理员将驳回申请，以确保接入控制网络的设备均有完善的维护信息。考虑到用户申请 IP 地址时经常无法确定将要接入的交换机名称和端口，系统提供了设备接入网络后根据采集的动态信息一键更新申请信息的功能。IP 地址申请的完整流程如图 4 所示。

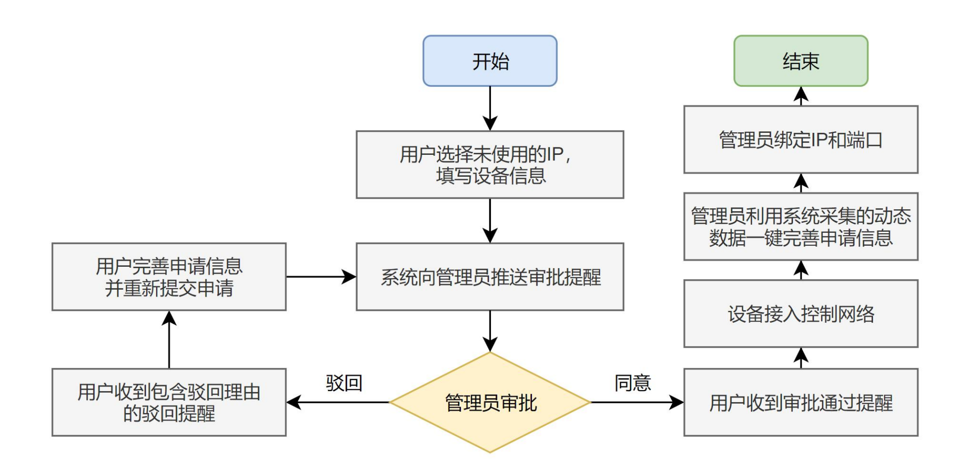



**图 4** IP 申请流程
**Fig.4** IP application process

## 2.2 数据库及界面设计

在数据存储设计方面，本文在 MongoDB 中设计了 6 个集合(collection)用于存储和管理与 IP 地址相关的静态数据。这些集合分别用于存储 IP 信息、申请信息、网段信息、位置信息、交换机信息及交换机端口信息，集合间通过共同的字段如 IP 地址、VLAN ID、设备连接的交换机 IP 和端口以及位置名称进行关联。以请求 IP 列表为例，集合间的关联关系如 5 所示。

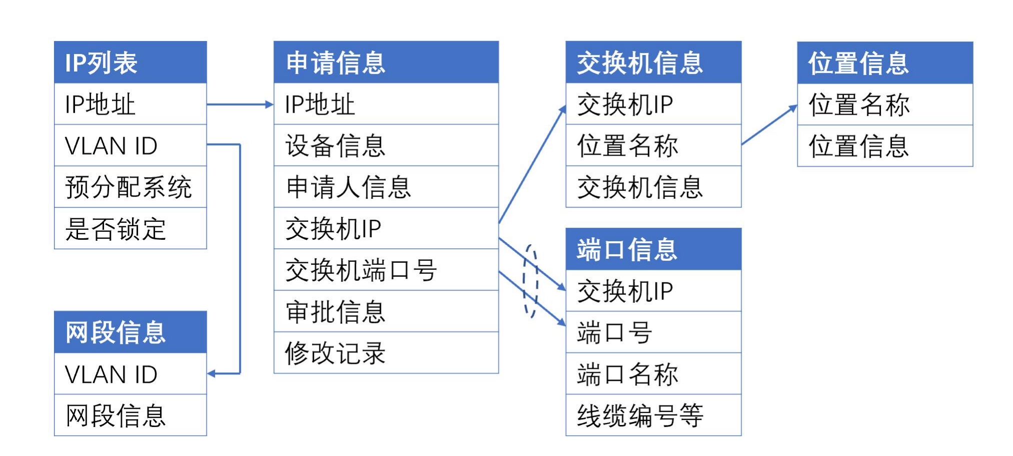


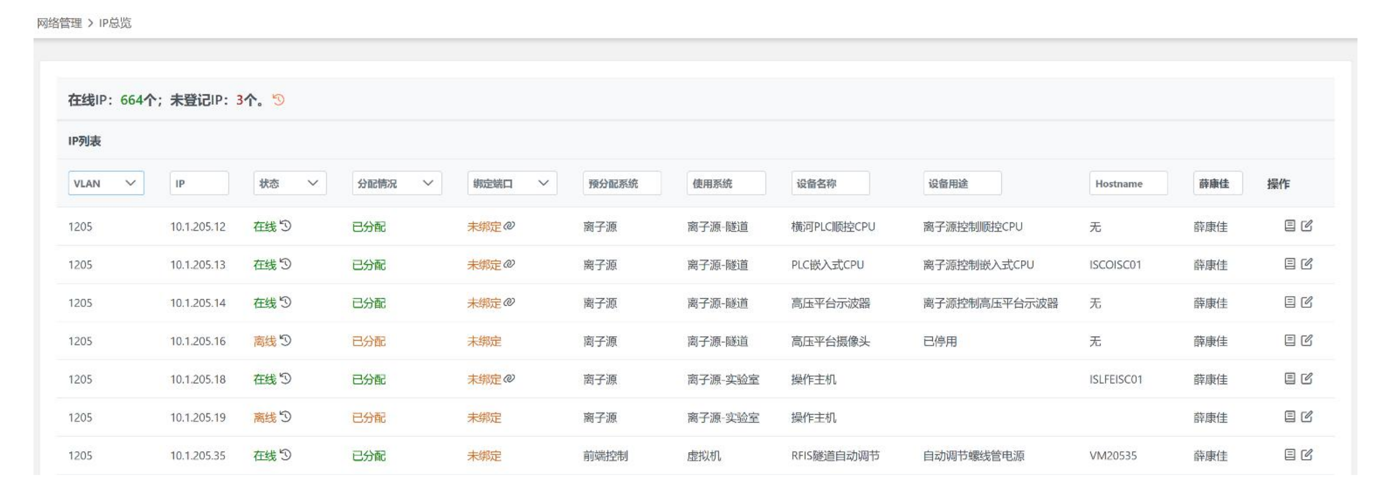

**图 5** IP 地址与静态信息关联关系
**Fig.5** Relationship between IP addresses and static information

本系统设计并实现了 7 个用户操作页面，包括 IP 总览、个人 IP 查询、交换机信息、端口信息、网段信息、管理员审批及数据导入页面。系统根据用户权限级别，精细划分了不同页面的访问与操作权限：管理员审批及数据导入界面仅供管理员使用，主要功能为审核用户申请和批量数据导入；个人 IP 查询页面限

定登录用户访问，用于查看和修改个人网络设备信息；IP 总览、交换机信息、端口信息和网段信息页面对所有用户开放，并按用户类型提供差异化功能，未登录用户可查询控制网络运维基本信息，已登录用户可提交 IP 地址使用申请，管理员则拥有修改任意 IP 信息和执行网络管理操作的最高权限。

用户访问 IP 总览页面时，后端服务程序按照图 5 的关联关系使用 MongoDB 的聚合方法(aggregate)返回控制网络可用的 IP 列表及相关信息。系统结合聚合管道中的$match 阶段实现按 VLAN、IP 地址、设备名称等多种属性进行检索[13]。IP 总览用户界面如图 6 所示。

通过结合网络信息和设备信息的设计，本系统不仅实现了 IP 地址的统一管理，还显著提高了网络信息的完整性和可追溯性。有效降低了 IP 冲突风险，同时为网络故障排查提供了便利，大大提升了 CSNS 加速器控制网络的管理效率和安全性。




| VLAN | IP | 状态 | 分配情况 | 绑定端口 | 预分配系统 | 使用系统 | 设备名称 | 设备用途 | Hostname | 薛康佳 | 操作 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1205 | 10.1.205.12 | 在线 | 已分配 | 未绑定 | 离子源 | 离子源-隧道 | 横河PLC顺控CPU | 离子源控制顺控CPU | 无 | 薛康佳 | |
| 1205 | 10.1.205.13 | 在线 | 已分配 | 未绑定 | 离子源 | 离子源-隧道 | PLC嵌入式CPU | 离子源控制嵌入式CPU | ISCOISC01 | 薛康佳 | |
| 1205 | 10.1.205.14 | 在线 | 已分配 | 未绑定 | 离子源 | 离子源-隧道 | 高压平台示波器 | 离子源控制高压平台示波器 | 无 | 薛康佳 | |
| 1205 | 10.1.205.16 | 离线 | 已分配 | 未绑定 | 离子源 | 离子源-隧道 | 高压平台摄像头 | 已停用 | 无 | 薛康佳 | |
| 1205 | 10.1.205.18 | 在线 | 已分配 | 未绑定 | 离子源 | 离子源-实验室 | 操作主机 | | ISLFEISC01 | 薛康佳 | |
| 1205 | 10.1.205.19 | 离线 | 已分配 | 未绑定 | 离子源 | 离子源-实验室 | 操作主机 | | | 薛康佳 | |
| 1205 | 10.1.205.35 | 在线 | 已分配 | 未绑定 | 前端控制 | 虚拟机 | RFIS隧道自动调节 | 自动调节螺线管电源 | VM20535 | 薛康佳 | |

**图 6** IP 信息总览页面
**Fig.6** IP information overview page

## 3 网络动态信息

在加速器网络管理中，需要查询的网络动态信息主要包括网络设备的在线信息和位置信息两部分。在线信息包括设备的在线状态、上线记录、离线记录以及 MAC 地址变化记录，这些信息有助于管理员判断设备是否接入控制网络以及是否发生设备更换。位置信息则指网络设备的物理位置，包括连接的交换机名称、端口及所在的设备厅。

市面上已有一些工具可采集网络动态信息，比如开源的网络监控软件 Netdisco、NeDi、LibreNMS 等[14]，以及商业的网络管理软件 eSight 等，这些工具能提供良好的设备发现与 IP 追踪功能，但在自定义数据采集方面不够灵活，且对网络设备的配置能力较为有限。

针对现有工具的局限性，本文开发了一套定制化的网络动态信息采集程序。定制化开发有几个关键优势：首先，定制化开发实现了与现有系统架构的无缝集成。本系统采集的网络动态信息需要与 IP 地址管理等其他功能模块协同工作，通过定制化开发，采集程序直接将数据存入 MongoDB 数据库，与其他模块共享统一的数据结构和访问接口，提升了各模块间的耦合度和数据一致性，为后续的数据分析和展示提供了坚持基础。其次，定制化开发提供了更为灵活的配置能力。例如在网络接入控制模块中，系统需要针对不同厂商的交换机实施差异化的配置策略。最后，定制化开发确保了系统的长期可维护性和可扩展性。在网络环境发生变化的情况下，系统能够快速响应新的需求，如增加对新型设备的支持、调整数据采集策略等。

### 3.1 在线信息

本文开发了两个模块来获取和维护设备的在线信息：ARP 表采集模块和 ARP 自动扫描模块。这两个模块协同工作，确保系统能够准确、高效地监控网络中设备的在线状态。以下将详细介绍模块的实现原理

和工作机制。

设备在线状态的获取方法主要有三种：一种常见方案是利用 ICMP 协议定期探测目标设备，该方法实现简单，但是会增加网络负载，且部分设备可能因安全策略禁用 ICMP 响应，导致数据不准确。方案二是基于网络流量进行分析，优点是无需主动探测，可减少网络干扰，但是实现复杂，需要部署专用硬件或软件，且难以检测到静默设备。方案三是基于核心交换机 ARP 表的监测方案，无需主动探测每个设备，避免了产生额外的网络流量，只需访问核心交换机，无需额外增加硬件，不受设备类型和操作系统限制。基于 ARP 表的监测方案的主要缺点是实时性受限于核心交换机中配置的 ARP 表老化时间，例如本文使用的核心交换机默认老化时间为 20 分钟，则系统获取 IP 在线状态的更新时间最快为 20 分钟。尽管如此，考虑到控制网络中设备状态变化的频率通常较低，且大多数设备一旦接入网络后会保持长期在线状态，这一更新周期仍能有效捕捉大部分设备状态的变化。因此，这一时间间隔能够满足控制网络管理的需求。

基于上述分析，本文采用基于 ARP 表的监测方案。该方案不仅具有广泛的适用性，能够适配各种加速器控制网络环境，还能在有效监控网络设备状态的同时最小化对网络性能的影响。

地址解析协议(ARP)是 TCP/IP 协议族中负责将 IP 地址映射到物理地址的关键协议[15]。当设备需要与其他设备通信时，通信过程根据目标设备所处位置遵循不同机制：对于同一网段内的通信，发送方首先查询本地 ARP 缓存；若无对应记录，则发送 ARP 请求广播，目标设备回应包含其 MAC 地址的 ARP 应答。对于跨网段通信，数据包将通过网关进行转发，设备与网关之间的 ARP 交互被记录在核心交换机中。这些交互过程使核心交换机能够在其 ARP 表中累积并维护一个包含网络中活跃设备的 IP-MAC 映射关系动态表，涵盖连接到各个网段的设备信息。

系统每 20 分钟执行一次核心交换机 ARP 表采集模块。该模块采用 Netmiko[16]库通过 SSH 协议连接核心交换机，执行查看 ARP 表项的命令，并通过 TextFSM 解析返回的文本[17]。当 IP 地址存在于 ARP 表且 MAC 地址不为 Incomplete 时，系统将该 IP 标记为在线状态，否则标记为离线。

为存储在线状态，本文设计了 3 个 MongoDB 集合：sw_core_arp、logs_ip_online 及 logs_mac_change。每次采集后，系统将在线 IP 及对应的 MAC 地址写入 sw_core_arp 集合，该集合中存储的 IP 地址即为当前在线 IP。同时，系统将当前采集的数据与上一次采集的数据进行比较，如果在线状态发生变化，即从在线转为离线或从离线转为在线，系统将变化情况写入 logs_ip_online 集合，如果 IP 地址映射的 MAC 地址发生变化，则将变化情况写入 logs_mac_change 集合。

为避免遗漏接入网络后无数据交互的静默设备，系统每天执行一次核心交换机 ARP 自动扫描模块，促使核心交换机能够在其动态 ARP 表中收录所有接入网络的设备信息。ARP 表采集方案如图 7 所示，程序依次在各网段执行 ARP 扫描任务，该任务用于触发核心交换机 ARP 自动扫描功能，即对接口 IP 地址所在网段的 IP 地址发送 ARP 请求报文，从而学习相应的 ARP 表项。当 ARP 表项到期后，交换机将启动重新学习机制。一旦设备信息被记录到核心交换机的 ARP 表中，只要该设备保持网络连接，交换机将通过周期性的 ARP 请求来维护和更新这些表项，因此无需频繁执行 ARP 自动扫描操作。考虑到 ARP 自动扫描会消耗核心交换机大量资源，扫描操作安排在网络相对空闲的凌晨时段执行。

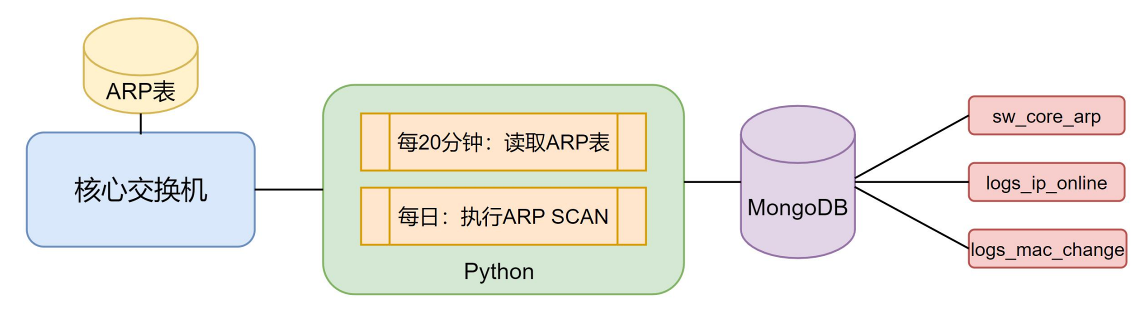



**图 7** ARP 表采集方案
**Fig.7** ARP table collection scheme

基于采集的 ARP 表数据，系统支持通过用户界面查询设备的实时在线状态、MAC 地址，以及在线/

离线记录和 MAC 地址变化记录，其中网络状态变化记录查询页面如图 8 所示。结合前文所述 IP 地址申请信息，在线状态可分为 4 种情况：在线、未分配但在线、已分配但离线、离线，其中未分配但在线属于异常状态，这说明有未经审批的设备自行接入了控制网络，在图 6 所示的 IP 总览页面中可通过状态和分配情况下拉框组合筛选这 4 种状态的 IP 列表。

基于核心交换机 ARP 表监测在线信息的方法，具有高效、低干扰和广泛适用性的特点，这种方法不仅能够识别正常的设备接入和断开，还能检测未经授权的设备接入，为网络安全管理提供了有力支持。

IP上线/离线记录　MAC地址变化记录

| IP地址 | 上线/离线 | MAC地址 | 时间 |
|---|---|---|---|
| 10.1.206.155 | 上线 | 80ae-5463-e1c5 | 2025-03-10 14:40 |
| 10.1.206.153 | 上线 | 80ae-5463-e105 | 2025-03-10 14:40 |
| 10.1.206.154 | 上线 | 80ae-5463-e1ff | 2025-03-10 14:40 |
| 10.1.202.63 | 上线 | 0040-9e04-1d5c | 2025-03-10 14:20 |
| 10.1.206.157 | 上线 | a8b1-3b71-c70e | 2025-03-10 13:40 |
| 10.1.206.157 | 离线 | | 2025-03-10 12:20 |

**图 8** 网络状态变化记录查询页面
**Fig.8** Network status change log query page

## 3.2 位置信息

大型加速器的网络设备分布广泛，遍及多个区域，涉及众多设备管理人员。由于加速器从建设到运维的周期通常较长，加之人员交接等因素，在过往的运行维护过程中，设备位置的准确定位成为一个常见挑战。为解决这一问题，本文通过采集网络动态信息并结合管理员录入的静态信息，实现了对任意网络设备位置的精确定位。该方法能够将设备位置精确到所在设备厅、所连接的交换机名称、端口号以及网线编号等，如图 9 所示。




**图 9** IP 详细信息页面
**Fig.9** IP detailed information page

本文开发了采集模块，每天自动采集核心交换机及所有接入交换机的接口信息和 MAC 地址表。通过分析交换机的接口信息及 MAC 地址表，系统能够获取每个交换机端口的连接状态、VLAN ID 以及所连接

设备的 MAC 地址。结合核心交换机的 ARP 表中所包含的 IP 和 MAC 地址映射关系，系统可以精确定位网络设备与特定交换机端口的连接关系。此外，由于管理员在维护交换机信息时已录入交换机所在设备厅以及每个交换机端口对应的网线编号，因此系统能够进一步确定网络设备的具体物理位置及其连接的网线编号。在 MongoDB 中 IP 地址通过动态信息关联位置的逻辑如图 10 所示，图中展示了系统中存储的两类数据：IP 列表、交换机信息和端口信息为管理员录入的静态信息，核心交换机 ARP 表、核心交换机接口信息和接入交换机 MAC 表为采集模块存储的动态信息。

这种动静态信息结合的方式确保了位置信息的准确性和实时性，解决了传统管理中设备位置难以追踪的问题。系统提供了从 IP 地址到具体物理位置的完整映射，提高了网络设备管理的准确性。

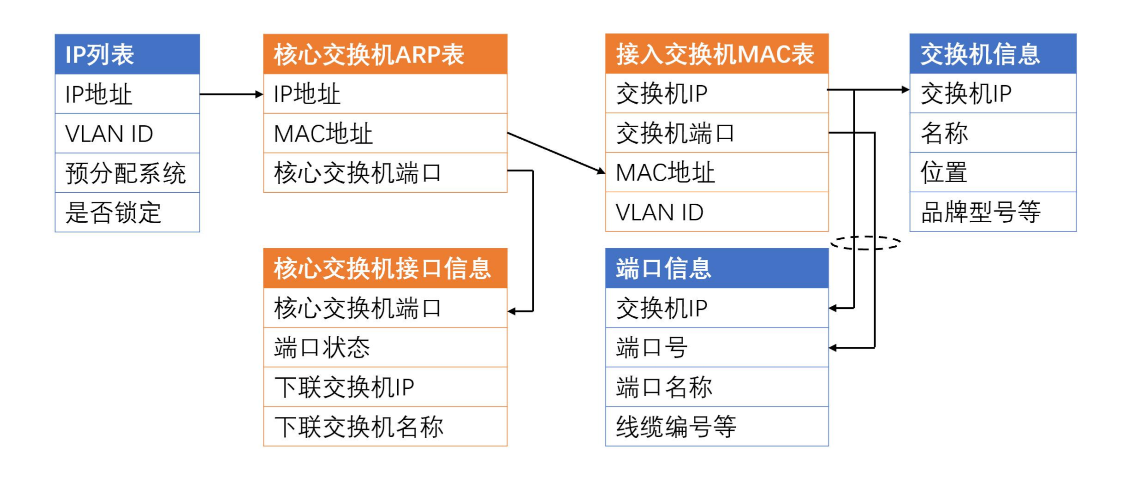



**图 10** IP 地址与动态信息关联关系

**Fig.10** Relationship between IP addresses and dynamic information

## 4 网络接入控制

### 4.1 网络接入控制方案设计

在常规的网络管理中，主要有 3 种网络接入控制的认证方式[18]：802.1X 认证、MAC 认证、Portal 认证。802.1X 是 IEEE 标准的基于端口的网络访问控制协议，主要用于有线和无线局域网环境中的身份认证。MAC 认证是一种基于端口和 MAC 地址对用户的网络访问权限进行控制的认证方法，它不需要用户安装任何客户端软件。Portal 认证通常也称为 Web 认证，是一种基于 Web 页面的认证方式。

然而，加速器控制网络中的大多数设备，如电源、示波器和串口服务器等，通常不具备采用 802.1X 认证或 Portal 认证的条件，因为这些设备无法安装认证软件或浏览器。考虑到加速器需要 24 小时不间断运行的特性，采用 MAC 认证方式在设备更换时需要重新认证，可能会影响加速器的正常运行。

针对这一特殊需求，本文设计了一种基于 IP 和端口绑定的接入控制方案。该方案的优势在于，设备更换时无需再次绑定，只需配置相同的 IP 地址即可接入网络，既保障了控制网络的安全性和可控性，又确保了加速器运行的连续性。

### 4.2 IP 和端口绑定的技术实现

IP 和端口绑定的实现方案与交换机硬件密切相关，不同厂商的设备在功能支持和配置方式上存在显著差异。为了应对这种多样性，本文开发了定制化的程序模块，以灵活适配多种厂商的设备。例如 CSNS 加速器控制网络目前采用了华为和锐捷两个厂商的接入交换机，需要实施差异化的配置策略。

对于华为交换机，本文利用了 IP Source Guard(IPSG)功能。IPSG 是一种基于二层接口的源 IP 地址过滤技术。通过在交换机接入用户侧的接口开启 IPSG 功能，交换机能够对进入接口的 IP 报文进行检查，丢弃非法主机的报文，从而有效阻止未经审批的设备接入控制网络。

而对于锐捷交换机，虽然也支持 IPSG 功能，但其 IPSG 规则要求同时绑定 IP、MAC 和端口，无法满

足仅绑定 IP 和端口的需求[19]。因此，本文采用了端口安全功能来实现 IP 和端口绑定。该功能通过限定允许进入交换机端口的源 IP 地址，实现限制未经审批设备接入控制网络。需要指出的是，华为交换机虽然也支持端口安全功能，但仅支持 MAC 和端口绑定，无法实现 IP 和端口绑定。

系统的 IP 和端口绑定流程如图 11 所示。为适应不同厂商设备的特性，本文开发了模块化的硬件交互组件，用于执行各类交换机操作。当管理员在用户界面提交 IP 和端口绑定请求时，系统首先将请求发送到 Node.js 后端服务；在 Node.js 服务中验证发起请求的用户是否具有管理员权限，通过验证后将请求转发到 Flask 服务；Flask 服务随即依据请求类型调用相应的硬件交互模块，该模块负责与相应的接入交换机建立连接并执行绑定操作；绑定操作完成后，硬件交互模块从交换机回读配置数据，并将这些数据写入 MongoDB 数据库；最后，将执行结果返回至用户界面，为管理员提供实时的操作反馈。

除 IP 和端口绑定功能外，本文还实现了端口使能管理和修改端口 VLAN ID 等功能。这些功能遵循相同的流程架构，仅在硬件交互模块的调用上有所不同，因此不再赘述。

通过这种创新的接入控制方案，本文有效解决了加速器控制网络中特殊设备的接入认证问题，提高了网络管理的灵活性和安全性，同时确保了加速器的持续稳定运行。

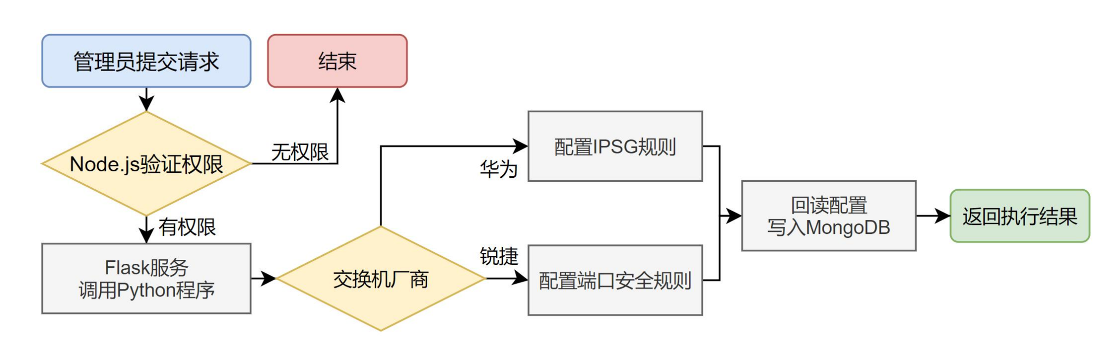



**图 11** IP 和端口绑定流程
**Fig.11** IP-port binding process

## 5 结语

本文针对大型加速器控制网络管理面临的挑战，为 CSNS-II 加速器设计并实现了一套全面的控制网络管理系统，实现了如下目标：（1）统一 IP 地址管理机制有效解决了网络资源冲突问题。本系统建立了 IP 地址统一申请、审批及分配机制，并详细记录了每个 IP 对应设备信息，为后续网络维护和故障排查提供了完整的基础数据支持。（2）关联网络动态信息显著提升了运维效率。本系统通过采集交换机动态信息，实现了设备在线状态的实时监控，并支持精确定位任意网络设备的物理位置，大幅提高了网络故障排查的效率。（3）IP 与端口绑定的网络接入控制机制有效增强了网络安全性。针对加速器控制网络的特殊需求，本系统实现了基于 IP 和端口绑定的接入控制方案，在保障网络安全的同时，确保了加速器的连续稳定运行。

本系统已在 CSNS 加速器控制网络中成功部署并稳定运行，有效解决了 IP 冲突、设备定位困难、网络接入无管控等长期存在的隐患。这不仅为 CSNS-II 网络管理奠定坚实基础，还有助于预防潜在网络风险，保障加速器安全稳定运行。此外，系统的设计思路和实现方法也为其他大规模加速器的网络管理提供了可复制的经验。

## 参考文献

1 王生，傅世年，屈化民，等. 中国散裂中子源强流质子加速器设计、研制及调试运行[J]. 原子能科学技术,2022,56(09):1747-1759. DOI: 10.7538/yzk.2022.youxian.0591.
Wang Sheng, Fu Shinian, Qu Huamin, et al. Development and Commissioning for High-intensity Proton Accelerator of China Spallation Neutron Source [J]. Atomic Energy Science and Technology,2022,56(09):1747-1759. DOI: 10.7538/yzk.2022.youxian.0591.

2 Zhang Y, Jin D, Zhu P, et al. The accelerator control system of CSNS[J]. Radiation Detection Technology and Methods, 2020, 4: 478-491. DOI: 10.1007/s41605-020-00203-y.

3 Geng Q, Huang X. VRRP load balance technology simulation practice based on GNS3[C]. MATEC Web of Conferences. EDP Sciences, 2018, 228: 03012.

4 贾娟,汪斌强,杨帅.一种基于 VRRP 的核心路由器高可用性方法研究与实现[J]. 电子技术应用,2007,33(2):110-112. DOI:10.3969/j.issn.0258-7998.2007.02.036.

5 Mehdizadeha A, Suinggia K, Mohammadpoorb M, et al. Virtual local area network (VLAN): Segmentation and security[C]. The Third International Conference on Computing Technology and Information Management (ICCTIM2017). 2017: 78-89.

6 Yu H, Qin T, Cui L, et al. Design of the network system for Hefei advanced light facility[J]. Radiation Detection Technology and Methods, 2025: 1-8. DOI: 10.1007/s41605-025-00541-9

7 Vue.js guide[EB/OL]. : https://vuejs.org/guide/introduction.html, 2025-3-10.

8 PrimeVue homepage[EB/OL]. : https://primevue.org, 2025-3-10.

9 Tilkov S, Vinoski S. Node. js: Using JavaScript to build high-performance network programs[J]. IEEE Internet Computing, 2010, 14(6): 80-83. DOI: 10.1109/MIC.2010.145.

10 Mazin A A, Abidin H Z, Mazalan L, et al. Network Automation Using Python Programming to Interact with Multiple Third-Party Network Devices[C] 2023 10th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). IEEE, 2023: 59-64. DOI: 10.1109/ICITACEE58587.2023.10277400.

11 Győrödi C, Győrödi R, Pecherle G, et al. A comparative study: MongoDB vs. MySQL[C] 2015 13th international conference on engineering of modern electric systems (EMES). IEEE, 2015: 1-6. DOI: 10.1109/EMES.2015.7158433.

12 Gu Y, Wang X, Shen S, et al. Analysis of data storage mechanism in NoSQL database MongoDB[C] 2015 IEEE International Conference on Consumer Electronics-Taiwan. IEEE, 2015: 70-71. DOI: 10.1109/ICCE-TW.2015.7217036

13 MongoDB manual[EB/OL]. : https://www.mongodb.com/docs/manual/aggregation/, 2025-3-17.

14 Qin T, Li C, Sun S A, et al. A device information-centered accelerator control network management system[J]. Radiation Detection Technology and Methods, 2024, 8(3): 1342-1358. DOI: 10.1007/s41605-024-00459-8.

15 Plummer D C.An ethernet address resolution protocol[EB/OL]. : https://www.rfc-editor.org/rfc/rfc826, 2025-3-17.

16 Netmiko homepage[EB/OL]. : https://ktbyers.github.io/netmiko/, 2025-3-17.

17 TextFSM homepage[EB/OL]. : https://github.com/google/textfsm, 2025-3-17.

18 华为交换机产品文档[EB/OL]. : https://support.huawei.com/hedex/hdx.do?docid=EDOC1100407964, 2025-3-24.

19 锐捷交换产品实施一本通[EB/OL]. : https://www.ruijie.com.cn/fw/wd/57629/, 2025-3-24.